CROPLAND CLASSIFICATION USING OPTICAL AND RADAR DATA

TEAM: AKASH A (191EC102), NAVRATAN (191EC133) AND ROHAN J (191EC147)

MENTOR: DR. RAGHAVENDRA B S

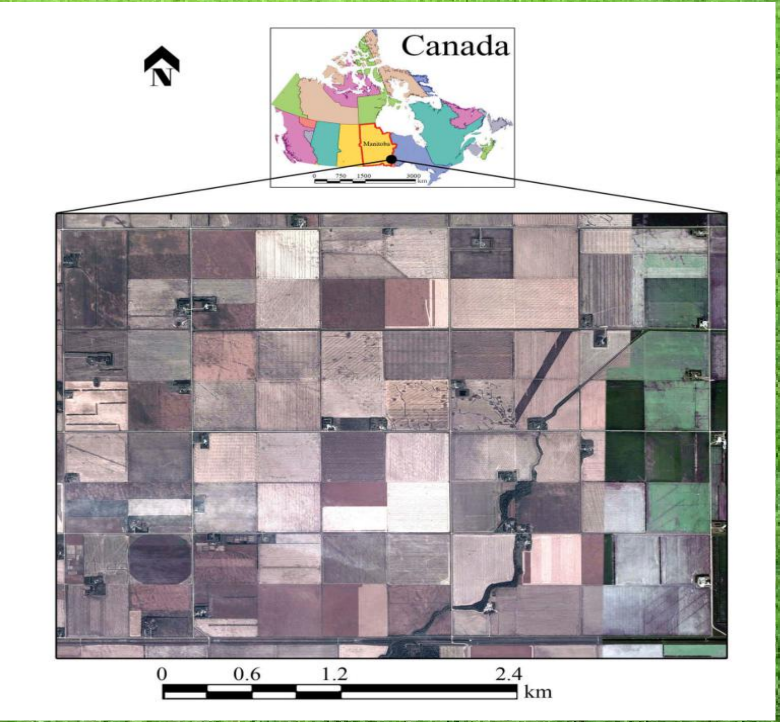
A BASIC IDEA

- Aim: To develop an ML model that can efficiently label croplands using data from remote sensing instruments

- Motivation: Possibility of use in agriculture planning and management activities at national and global scales

- Area under Study: Southwest district of Winnipeg, Manitoba, Canada - Covered by various annual crops

- Seven different types of crops - Broadleaf, Soybeans, Canola, Wheat, Peas, Corn and Oats



UNDERSTANDING THE DATASET


```
1,-13.559,-21.407,-11.404,-15.248,-11.923,-15.291,-2.1548,-7.8474,-10.002,0.04239,3.3253,3.3677,0.35631,0.05849,0.5852,0.2415,0.51934,0.23916,-0.62424,-0.81493
1,-12.802,-20.335,-10.399,-14.132,-11.096,-14.361,-2.4039,-7.533,-9.9369,0.22842,3.036,3.2644,0.34295,0.060525,0.59652,0.25249,0.50796,0.23955,-0.57229,-0.7485
1,-12.431,-19.902,-10.074,-13.598,-10.829,-14.048,-2.3566,-7.4717,-9.8283,0.44978,2.7687,3.2185,0.34489,0.061731,0.59338,0.26362,0.4987,0.23768,-0.53347,-0.702
1,-12.689,-19.529,-10.028,-13.35,-11.056,-14.014,-2.6611,-6.8396,-9.5006,0.66378,2.2942,2.958,0.3276,0.067825,0.60457,0.28135,0.47717,0.24148,-0.50405,-0.66998
1,-12.686,-19.278,-9.8185,-13.108,-10.932,-13.939,-2.8675,-6.5919,-9.4594,0.83143,2.1756,3.007,0.31701,0.069483,0.61351,0.28768,0.47476,0.23756,-0.50314,-0.682
1,-12.478,-19.034,-9.6201,-12.888,-10.761,-13.687,-2.8583,-6.5555,-9.4138,0.79854,2.1276,2.9261,0.31723,0.070118,0.61265,0.28867,0.47115,0.24018,-0.49984,-0.69
1,-12.459,-19.019,-9.3854,-12.766,-10.664,-13.393,-3.0739,-6.5599,-9.6338,0.62733,2.1023,2.7297,0.30766,0.067934,0.6244,0.28668,0.46519,0.24812,-0.45483,-0.646
1,-12.721,-19.057,-9.4054,-12.887,-10.797,-13.384,-3.3153,-6.3366,-9.6519,0.49645,2.0904,2.5868,0.29604,0.068815,0.63515,0.28488,0.461,0.25411,-0.42858,-0.6054
1,-12.777,-19.202,-9.2785,-12.905,-10.762,-13.213,-3.4982,-6.4252,-9.9234,0.3078,2.1437,2.4515,0.28855,0.065722,0.64572,0.28013,0.45891,0.26096,-0.39564,-0.585
1,-12.804,-19.273,-9.2722,-12.937,-10.802,-13.143,-3.5322,-6.4682,-10,0.2059,2.1353,2.3412,0.28728,0.064787,0.64793,0.27865,0.45561,0.26575,-0.37654,-0.55633,
1,-12.631,-19.305,-9.1779,-12.851,-10.717,-12.968,-3.4531,-6.6742,-10.127,0.11687,2.1342,2.2511,0.29157,0.062707,0.64572,0.27716,0.45305,0.2698,-0.35872,-0.549
1,-12.526,-19.121,-9.2445,-12.8,-10.686,-13.074,-3.2812,-6.5948,-9.876,0.27402,2.1146,2.3887,0.29871,0.065428,0.63586,0.28042,0.45632,0.26327,-0.38911,-0.62218
1,-12.192,-18.994,-9.0455,-12.623,-10.381,-12.899,-3.1461,-6.8028,-9.9489,0.27559,2.2417,2.5173,0.30559,0.063806,0.6306,0.27669,0.46363,0.25968,-0.39604,-0.630
1,-11.973,-18.769,-8.8934,-12.557,-10.112,-12.771,-3.0795,-6.7958,-9.8753,0.21398,2.4454,2.6594,0.30852,0.064521,0.62696,0.26969,0.47359,0.25672,-0.38742,-0.58
1,-11.782,-18.758,-8.7488,-12.405,-9.9305,-12.679,-3.033,-6.9759,-10.009,0.27384,2.4748,2.7487,0.31142,0.062482,0.6261,0.26977,0.47695,0.25328,-0.38286,-0.5970
1,-11.961,-18.806,-8.7712,-12.504,-9.9391,-12.845,-3.1901,-6.845,-10.035,0.34063,2.5654,2.906,0.30383,0.062825,0.63334,0.26811,0.48401,0.24788,-0.39594,-0.6254
1,-12.03,-18.986,-8.8136,-12.614,-9.9955,-12.856,-3.2168,-6.9559,-10.173,0.24233,2.618,2.8604,0.30313,0.061099,0.63577,0.26504,0.4843,0.25066,-0.39173,-0.59243
1,-12.239,-18.905,-8.8095,-12.736,-10.005,-12.925,-3.429,-6.666,-10.095,0.18876,2.7312,2.92,0.29258,0.063044,0.64438,0.26089,0.48931,0.2498,-0.3755,-0.56765,
1,-12.402,-18.992,-8.9772,-12.82,-10.193,-13.117,-3.4253,-6.5895,-10.015,0.29631,2.6271,2.9234,0.29241,0.064127,0.64346,0.26559,0.48633,0.24808,-0.38812,-0.592
1,-12.653,-18.936,-9.1522,-12.99,-10.417,-13.228,-3.5005,-6.2833,-9.7838,0.23772,2.5728,2.8105,0.28783,0.067733,0.64444,0.26631,0.48157,0.25212,-0.38423,-0.549
1,-12.725,-19.009,-9.1793,-13.038,-10.488,-13.215,-3.5459,-6.2839,-9.8298,0.17695,2.5497,2.7266,0.28589,0.067269,0.64684,0.26604,0.47854,0.25542,-0.363,-0.5144
1,-12.776,-19.015,-9.1837,-13.088,-10.47,-13.265,-3.5919,-6.2393,-9.8312,0.17653,2.6184,2.7949,0.28374,0.067452,0.64881,0.26402,0.48248,0.2535,-0.35577,-0.5118
1,-12.879,-19.202,-9.2825,-13.068,-10.665,-13.337,-3.5961,-6.3233,-9.9194,0.26891,2.4022,2.6711,0.28393,0.066204,0.64986,0.27184,0.47264,0.25552,-0.35706,-0.52
1,-13.036,-19.303,-9.2518,-13,-10.784,-13.313,-3.7843,-6.2666,-10.051,0.31314,2.2158,2.529,0.27576,0.065142,0.6591,0.27808,0.46318,0.25874,-0.35658,-0.51447,
1,-13.475,-19.558,-9.4965,-13.164,-11.197,-13.553,-3.9782,-6.0835,-10.062,0.38866,1.9669,2.3556,0.26697,0.065783,0.66725,0.28676,0.45103,0.26221,-0.35943,-0.51
1,-14.145,-20.116,-10,-13.624,-11.791,-14.131,-4.1446,-5.9707,-10.115,0.50748,1.8327,2.3401,0.25975,0.065688,0.67456,0.29285,0.44659,0.26055,-0.36176,-0.49338,
1,-14.879,-20.842,-10.368,-14.1,-12.283,-14.609,-4.5111,-5.9629,-10.474,0.50869,1.8172,2.3259,0.24516,0.062111,0.69273,0.29334,0.44575,0.26092,-0.37849,-0.5089
1,-12.255,-18.035,-8.3644,-12.216,-9.8822,-12.353,-3.8906,-5.7801,-9.6707,0.13737,2.3336,2.471,0.26928,0.071152,0.65957,0.27172,0.46503,0.26326,-0.38486,-0.624
1,-11.98,-17.953,-8.474,-12.225,-9.8602,-12.341,-3.5064,-5.9724,-9.4789,0.11645,2.3648,2.4813,0.28614,0.072332,0.64153,0.27047,0.46622,0.26331,-0.4091,-0.66032
1,-12.089,-18.048,-8.7878,-12.598,-10.016,-12.591,-3.3013,-5.9589,-9.2603,-0.0070438,2.5825,2.5755,0.29479,0.074753,0.63045,0.26219,0.47519,0.26262,-0.44085,-
1,-11.873,-18.149,-8.9818,-12.786,-9.99,-12.691,-2.8913,-6.2762,-9.1675,-0.095092,2.7958,2.7007,0.3143,0.074085,0.61161,0.25473,0.48491,0.26037,-0.48928,-0.683
1,-11.589,-18.288,-9.2368,-12.921,-9.9906,-12.827,-2.3519,-6.6994,-9.0513,-0.094002,2.9305,2.8365,0.34101,0.072916,0.58608,0.25091,0.49269,0.2564,-0.5293,-0.74
1,-11.535,-18.405,-9.4756,-12.97,-10.077,-13.116,-2.059,-6.8702,-8.9292,0.14611,2.8936,3.0397,0.3556,0.073104,0.5713,0.2555,0.49745,0.24705,-0.54929,-0.76986,
1,-11.393,-18.468,-9.5114,-13.046,-9.9205,-13.256,-1.8821,-7.0742,-8.9563,0.20945,3.1256,3.335,0.36515,0.071623,0.56322,0.24958,0.51259,0.23783,-0.58269,-0.808
1,-11.168,-18.467,-9.5521,-13.015,-9.7848,-13.327,-1.616,-7.2992,-8.9152,0.31187,3.2301,3.542,0.37921,0.070626,0.55016,0.24786,0.52146,0.23068,-0.61299,-0.8544
1,-11.186,-18.414,-9.7536,-13.162,-9.9048,-13.386,-1.432,-7.2283,-8.6602,0.22455,3.2571,3.4816,0.38761,0.073379,0.53901,0.24591,0.52057,0.23352,-0.62418,-0.855
1,-11.208,-18.479,-9.7307,-13.249,-9.8712,-13.376,-1.4778,-7.2703,-8.748,0.12653,3.3779,3.5044,0.38568,0.07231,0.54201,0.24108,0.52476,0.23416,-0.63095,-0.8776
1,-11.195,-18.729,-9.8722,-13.382,-9.8773,-13.608,-1.323,-7.5338,-8.8568,0.2262,3.5049,3.7311,0.39486,0.069672,0.53547,0.23863,0.53484,0.22652,-0.7022,-0.93316
1,-11.265,-18.754,-9.9805,-13.467,-10.015,-13.581,-1.2848,-7.4883,-8.7731,0.11378,3.4521,3.5658,0.39642,0.070685,0.53289,0.23876,0.52865,0.23259,-0.70459,-0.91
1,-11.392,-18.769,-9.7879,-13.384,-9.9419,-13.602,-1.6042,-7.3769,-8.9811,0.21763,3.4424,3.66,0.38026,0.069565,0.55017,0.24037,0.53102,0.22862,-0.69031,-0.8931
1,-11.313,-18.982,-9.5346,-13.237,-9.7508,-13.516,-1.7784,-7.6688,-9.4471,0.2786,3.4864,3.765,0.37354,0.063894,0.56257,0.23983,0.53524,0.22493,-0.72655,-0.9560
1,-10.973,-18.855,-8.9881,-12.868,-9.2549,-13.117,-1.9853,-7.882,-9.8673,0.24868,3.613,3.8617,0.36465,0.059384,0.57597,0.23573,0.54165,0.22261,-0.6611,-0.85704
1,-10.671,-18.742,-8.5504,-12.546,-8.8781,-12.713,-2.121,-8.0703,-10.191,0.16741,3.6678,3.8353,0.35899,0.055982,0.58503,0.23315,0.54252,0.22433,-0.58106,-0.766
```


- ❖ Dataset consists of bitemporal optical and RADAR features
- ❖ RapidEye Satellites - Collected optical features from five spectral bands - R, G, B, NIR and RE
- ❖ UAVSAR - airborne SAR sensor operating in full polarization mode collected radar features
- ❖ Each row corresponds to a single pixel of data; First column represents label followed by 174 features

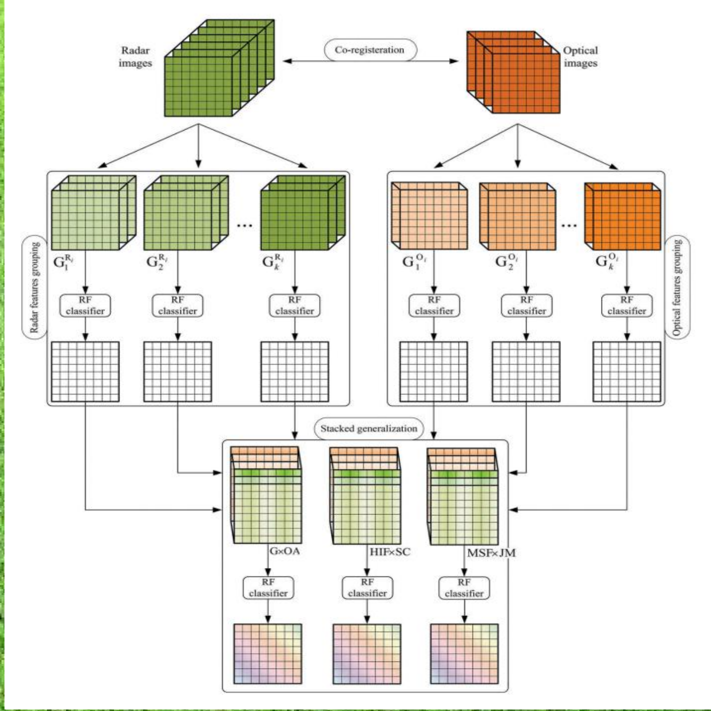
PROPOSED CROP MAPPING FRAMEWORK

Feature Extraction and Grouping

Classification of Each group

Stacking the Results of Each Group

Final Classification



STEP 1: FEATURE EXTRACTION AND GROUPING

RADAR FEATURES

Features and formulas (G: groups, R: radar, i = 1 to 7)

$G_1^{R_i}$	$\sigma_{hh} = 10\log_{10}\lvert S_{hh}\rvert^2, \sigma_{hv} = 10\log_{10}\lvert S_{hv}\rvert^2, \sigma_{vv} = 10\log_{10}\lvert S_{vv}\rvert^2, \sigma_{rr} = 10\log_{10}\lvert S_{rr}\rvert^2, \sigma_{rl} = 10\log_{10}\lvert S_{rl}\rvert^2, \sigma_{ll} = 10\log_{10}\lvert S_{ll}\rvert^2$ {h : horizontal, v : vertical, r : right – handed, l : left – handed}
$G_2^{R_i}$	$R_{hhvv} = 10\log_{10}(\lvert S_{hh}\rvert^2/\lvert S_{vv}\rvert^2), R_{hvhh} = 10\log_{10}(\lvert S_{hv}\rvert^2/\lvert S_{hh}\rvert^2), R_{hvvv} = 10\log_{10}(\lvert S_{hv}\rvert^2/\lvert S_{vv}\rvert^2),$ $R_{rrll} = 10\log_{10}(\lvert S_{rr}\rvert^2/\lvert S_{ll}\rvert^2), R_{rlrr} = 10\log_{10}(\lvert S_{rl}\rvert^2/\lvert S_{rr}\rvert^2), R_{rlll} = 10\log_{10}(\lvert S_{rl}\rvert^2/\lvert S_{ll}\rvert^2)$
$G_3^{R_i}$	$R_{hh} = 10\log_{10}(\lvert S_{hh}\rvert^2/\text{span}), R_{hv} = 10\log_{10}(\lvert S_{hv}\rvert^2/\text{span}), R_{vv} = 10\log_{10}(\lvert S_{vv}\rvert^2/\text{span}),$ $R_{rr} = 10\log_{10}(\lvert S_{rr}\rvert^2/\text{span}), R_{rl} = 10\log_{10}(\lvert S_{rl}\rvert^2/\text{span}), R_{ll} = 10\log_{10}(\lvert S_{ll}\rvert^2/\text{span})$ $\{\text{span} = \lvert S_{hh}\rvert^2 + 2\lvert S_{hv}\rvert^2 + \lvert S_{vv}\rvert^2 = \lvert S_{rr}\rvert^2 + 2\lvert S_{rl}\rvert^2 + \lvert S_{ll}\rvert^2\}$
$G_4^{R_i}$	$\rho_{hhvv} = \left\lvert \frac{S_{hh}.S_{vv}^*}{\sqrt{(S_{hh}.S_{hh}^*)(S_{vv}.S_{vv}^*)}} \right\rvert, \rho_{hvhh} = \left\lvert \frac{S_{hv}.S_{hh}^*}{\sqrt{(S_{hv}.S_{hv}^*)(S_{hh}.S_{hh}^*)}} \right\rvert, \rho_{hvvv} = \left\lvert \frac{S_{hv}.S_{vv}^*}{\sqrt{(S_{hv}.S_{hv}^*)(S_{vv}.S_{vv}^*)}} \right\rvert;$ $\rho_{rrll} = \left\lvert \frac{S_{rr}.S_{ll}^*}{\sqrt{(S_{rr}.S_{rr}^*)(S_{ll}.S_{ll}^*)}} \right\rvert, \rho_{rlrr} = \left\lvert \frac{S_{rl}.S_{rr}^*}{\sqrt{(S_{rl}.S_{rl}^*)(S_{rr}.S_{rr}^*)}} \right\rvert, \rho_{rlll} = \left\lvert \frac{S_{rl}S_{ll}^*}{\sqrt{(S_{rl}S_{rl}^*)(S_{ll}.S_{ll}^*)}} \right\rvert$ {* : complex conjugate, . : vector dot product}
$G_5^{R_i}$	$\lambda_1, \lambda_2, \lambda_3, \quad H = -\sum_{i=1}^3 p_i \log_3 p_i, A = (\lambda_2 - \lambda_3)/(\lambda_2 + \lambda_3), \bar{\alpha} = \sum_{i=1}^3 p_i \alpha_i \qquad \{p_i = \lambda_i / \sum_{k=1}^3 \lambda_k\}$ $HA, H(1-A), (1-H)A, (1-H)(1-A), \quad \psi = \min(\lambda_1, \lambda_2, \lambda_3)/(\lambda_1 + \lambda_2 + \lambda_3), \text{RVI} = 4\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$
$G_6^{R_i}$	$\lvert\alpha\rvert^2 = \frac{\sqrt{2}}{2}\lvert S_{hh} + S_{vv}\rvert^2, \lvert\beta\rvert^2 = \frac{\sqrt{2}}{2}\lvert S_{hh} - S_{vv}\rvert^2, \lvert\gamma\rvert^2 = 2\lvert S_{hv}\rvert^2, \lvert k_s\rvert^2 = \lvert S_{rl}\rvert^2, \lvert k_d\rvert^2 = \min(\lvert S_{rr}\rvert^2, \lvert S_{ll}\rvert^2), \lvert k_h\rvert^2 = \text{abs}(\lvert S_{rr}\rvert^2 - \lvert S_{ll}\rvert^2)$ {s : surface scattering, d : double – bounce scattering, h : helix scattering}
$G_7^{R_i}$	$P_s = f_s(1 + \lvert\beta\rvert^2), P_d = f_d(1 + \lvert\alpha\rvert^2), P_v = f_v, P_s^Y = f_s(1 + \lvert\beta\rvert^2), P_d^Y = f_d(1 + \lvert\alpha\rvert^2), P_v^Y = f_v, P_c^Y = f_c$ {v : volume scattering, c : helix scattering}

OPTICAL FEATURES

Features and formulas (G: groups, O: optical, i = 1 to 5)

$G_1^{O_i}$	R_B: 440 – 510nm, R_G: 520 – 590nm, R_R: 630 – 685nm, R_{RE}: 690 – 730nm, R_{NIR}: 760 – 850nm R: Reflectance
$G_2^{O_i}$	$\text{NDVI} = (R_{NIR} - R_R)/(R_{NIR} + R_R)$ $\text{SR} = R_{NIR}/R_R$ $\text{RGRI} = R_G/R_R$ $\text{EVI} = 2.5(R_{NIR} - R_R)/(R_{NIR} + 6R_R - 7.5R_B + 1)$ $\text{ARVI} = (R_{NIR} - (2R_R - R_B))/(R_{NIR} + (2R_R - R_B))$ $\text{SAVI} = (1 + 0.5)(R_{NIR} - R_R)/(R_{NIR} + R_R + 0.5)$ $\text{NDGI} = (R_G - R_R)/(R_G + R_R)$ $\text{gNDVI} = (R_{NIR} - R_G)/(R_{NIR} + R_G)$ $\text{MTVI2} = 1.5(1.2(R_{NIR} - R_G) - 2.5(R_R - R_G))/\sqrt{(2R_{NIR} + 1)^2 - (6R_{NIR} - 5\sqrt{R_R}) - 0.5}$
$G_3^{O_i}$	$\text{NDVIre} = (R_{NIR} - R_{RE})/(R_{NIR} + R_{RE})$ $\text{SRre} = R_{NIR}/R_{RE}$ $\text{NDGIre} = (R_G - R_{RE})/(R_G + R_{RE})$ $\text{RTVIcore} = 100(R_{NIR} - R_{RE}) - 10(R_{NIR} - R_G)$ $\text{RNDVI} = (R_{RE} - R_R)/(R_{RE} + R_R)$ $\text{TCARI} = 3((R_{RE} - R_R) - 0.2(R_{RE} - R_G)(R_{RE}/R_R))$ $\text{TVI} = 0.5(120(R_{RE} - R_G) - 200(R_R - R_G))$ $\text{PRI2} = R_{RE}/R_R$
$G_4^{O_i}$	$\mu_{PC1}, \sigma_{PC1}, \text{HOM}_{PC1}, \text{CON}_{PC1}, \text{DIS}_{PC1}, H_{PC1}, \text{ASM}_{PC1}, \text{COR}_{PC1}$ from GLCM of PC1
$G_5^{O_i}$	$\mu_{PC2}, \sigma_{PC2}, \text{HOM}_{PC2}, \text{CON}_{PC2}, \text{DIS}_{PC2}, H_{PC2}, \text{ASM}_{PC2}, \text{COR}_{PC2}$ from GLCM of PC2

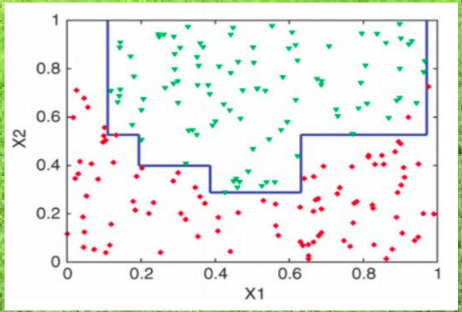
STEP 2: TRAINING EACH OF THE FEATURE GROUPS

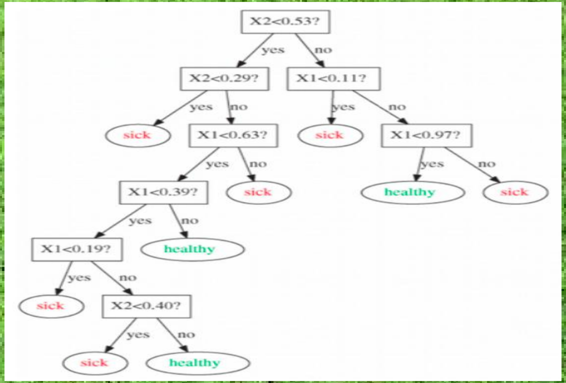
- Random Forest (RF) Classifier was selected owing to its stable nature and research support for current application

- RF Classifier: - A classifier based on decision tree classification

- Decision Tree: - A tree based ML model; Uses a series of questions to come up with the hypothesis function

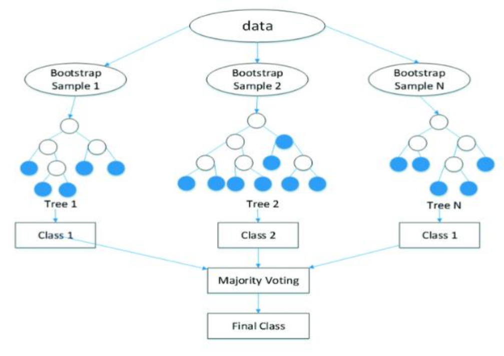
- Disadvantage of Decision Trees: - Weak classifier, often results in overfitting the training set

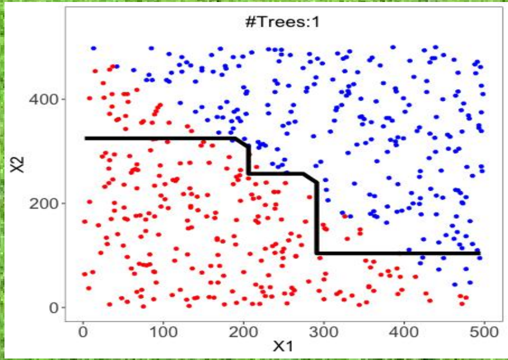


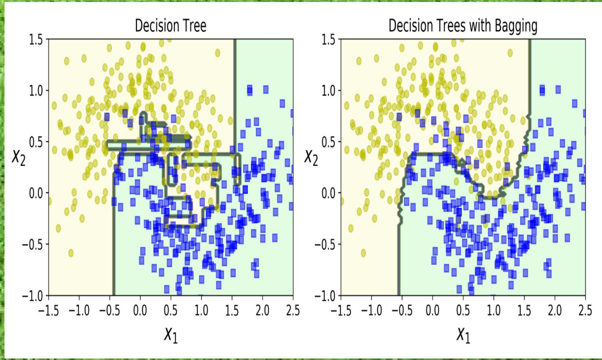


STEP 2 CONTD. : THE RANDOM FOREST CLASSIFIER

- Based on ensemble learning: - Combines results of several weak classifiers to get a stronger classifier
- Optimal number of decision trees used - Each given a subset of the training set and a subset of the features by bootstrapping the training set (Number of features given to each tree is important)
- Majority voting used on prediction of each decision tree to predict the result







STEPS 3 AND 4: STACKING AND FINAL CLASSIFIER

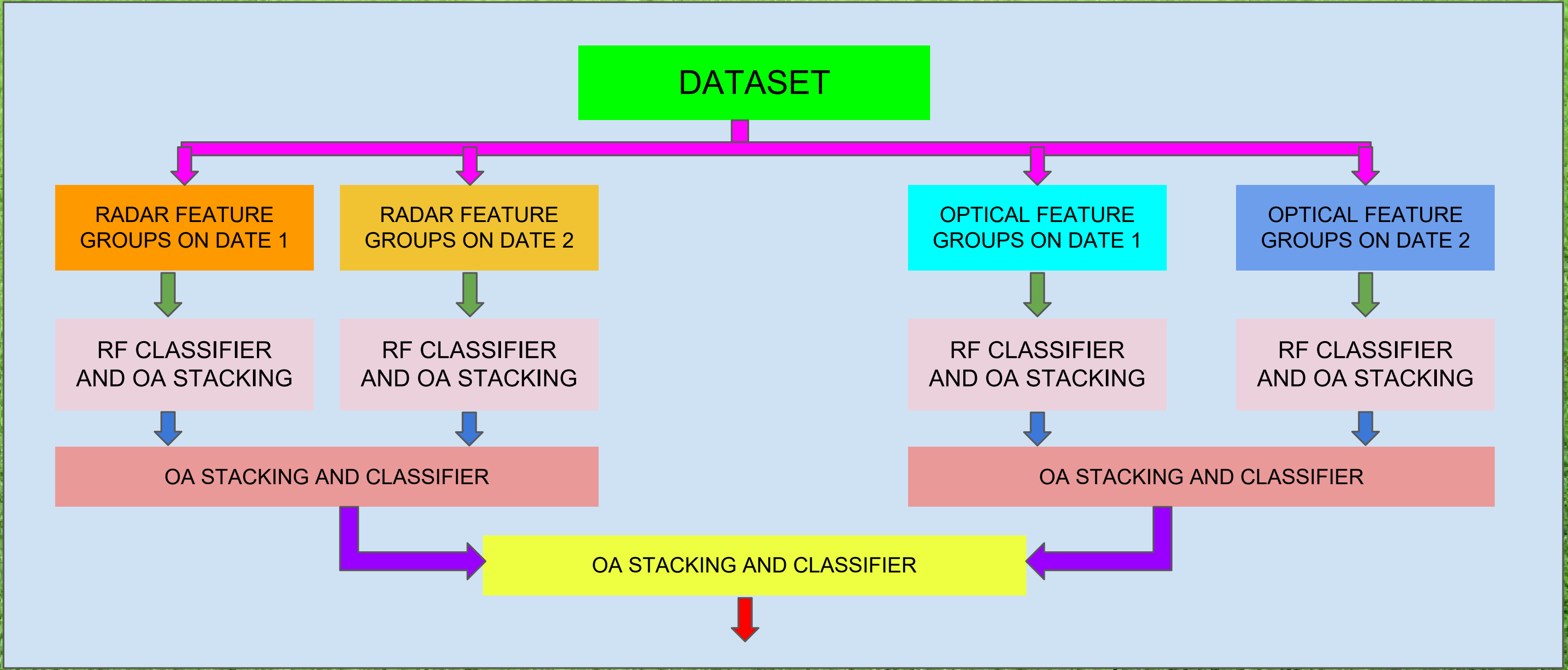
- Traditional approaches for stacking include using max voting on predictions of each classifier or using another ML model to select more important features. Three optimized approaches were suggested. Another RF classifier was used after the stacking.

- Approach 1 - Multiply the predictions on the training of each group by the Overall Accuracy (OA) of that group and then use it for next phase of training - more accurate groups get greater weight

- Approach 2 - Select a certain number of highest important features of each group during classification and use these for the next stage after multiplying by the importance scores and stacking

- Approach 3 - Select the most separable features of each group using Jeffries-Matusita (JM) distance and use these features for the next stage of classification

IMPLEMENTATION



CODE SEGMENT FOR RF CLASSIFICATION

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

Xtrain_group1_radar_date1 = balanced_training_data[:,[1, 2, 3, 4, 5, 6]]
classifier_radar1_date1 = RandomForestClassifier(n_estimators = 500, max_features = 'sqrt')
classifier_radar1_date1.fit(Xtrain_group1_radar_date1, y_train)
X_test_radar1_date1 = data[:, [1, 2, 3, 4, 5, 6]]
y_test = data[:, 0]
y_predicted_radar1_date1 = classifier_radar1_date1.predict(X_test_radar1_date1)
OA_group1_radar_date1 = accuracy_score(y_test, y_predicted_radar1_date1)
print('Overall Accuracy of Radar Group 1 on date 1: %f\n'%(OA_group1_radar_date1))

y_predict_train_group1_radar_date1 = classifier_radar1_date1.predict(Xtrain_group1_radar_date1)
next_group1_date1_radar = y_predict_train_group1_radar_date1*OA_group1_radar_date1
next_group1_date1_radar = np.reshape(next_group1_date1_radar, (-1, 1))
```

Class	Corn	Peas	Canola	Soybeans	Oats	Wheat	Broadleaf
No. of samples	39162	3598	75673	74067	47117	85074	1143

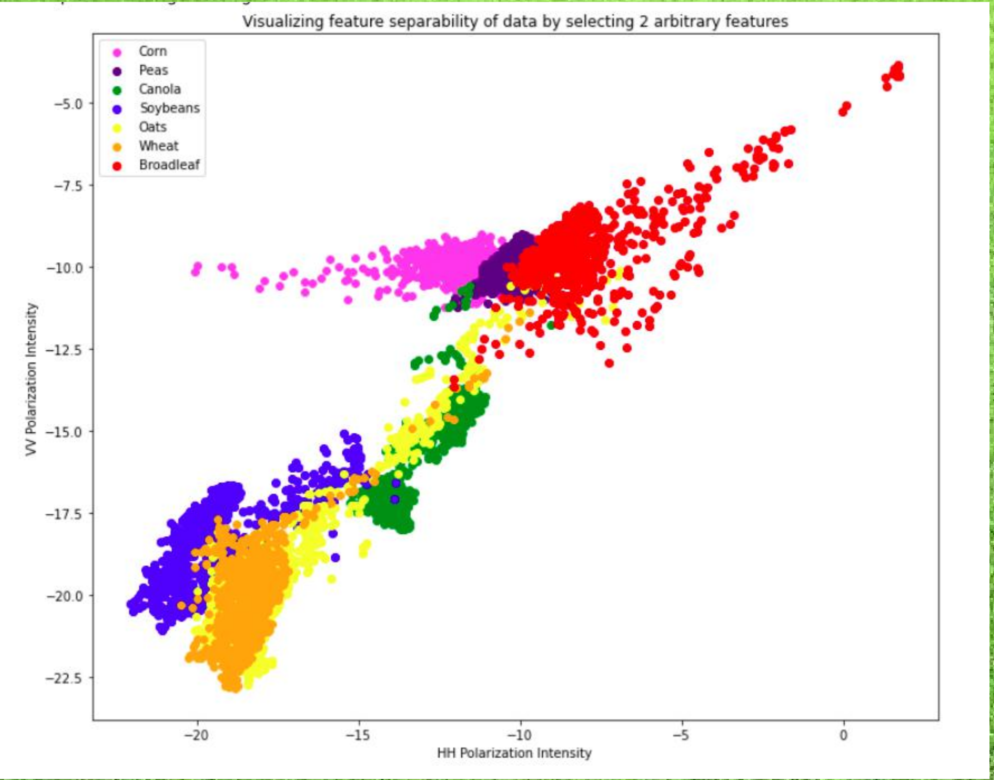
CODE SEGMENT FOR STACKING

```
Xtrain_radar_date1 = np.hstack((next_group1_date1_radar, next_group2_date1_radar,
next_group3_date1_radar, next_group4_date1_radar, next_group5_date1_radar, next_group6_date1_radar,
next_group7_date1_radar))

classifier_radar_date1 = RandomForestClassifier(n_estimators = 500, max_features = 'sqrt')
classifier_radar_date1.fit(Xtrain_radar_date1, y_train)

X_test_radar_date1 = np.hstack((y_predicted_radar1_date1*OA_group1_radar_date1,
y_predicted_radar2_date1*OA_group2_radar_date1, y_predicted_radar3_date1*OA_group3_radar_date1,
y_predicted_radar4_date1*OA_group4_radar_date1, y_predicted_radar5_date1*OA_group5_radar_date1,
y_predicted_radar6_date1*OA_group6_radar_date1, y_predicted_radar7_date1*OA_group7_radar_date1))
X_test_radar_date1 = np.transpose(np.reshape(X_test_radar_date1, (7, -1)))
y_predicted_radar_date1 = classifier_radar_date1.predict(X_test_radar_date1)
OA_radar_date1 = accuracy_score(y_test, y_predicted_radar1_date1)
print('Overall Accuracy of radar data on date 1 : %f\n'%(OA_radar_date1))
```

RESULTS



RADAR GROUPS FOR DATE 1	OVERALL ACCURACY
G1	60.33%
G2	52.10%
G3	51.28%
G4	49.59%
G5	61.88%
G6	50.09%
G7	50.47%

RESULTS - CONTD.

OPTICAL GROUPS FOR DATE 1	OVERALL ACCURACY
G1	66.67%
G2	59.53%
G3	65.67%
G4	45.32%
G5	38.95%

RADAR GROUPS FOR DATE 2	OVERALL ACCURACY
G1	75.24%
G2	48.05%
G3	45.68%
G4	37.48%
G5	75.96%
G6	63.58%
G7	66.97%

OPTICAL GROUPS FOR DATE 2	OVERALL ACCURACY
G1	66.28%
G2	40.33%
G3	56.25%
G4	37.55%
G5	32.05%

RESULTS - CONTD.

STACKED FEATURES	OVERALL ACCURACY
RADAR FEATURES - DATE 1	60.33%
RADAR FEATURES - DATE 2	75.24%
RADAR FEATURES	70.04%
OPTICAL FEATURES - DATE 1	66.67%
OPTICAL FEATURES - DATE 2	66.28%
OPTICAL FEATURES	59.33%
RADAR & OPTICAL FEATURES	64.03%

- Radar features were observed to offer a much higher overall accuracy as compared to optical feature groups
- Each group was found to have a higher accuracy than a random classifier which verifies that the model has learnt to classify
- Overall accuracy obtained was 64.03% which ranks on an average scale for a multi-class classification problem

THE WAY FORWARD...

- ❖ Attempt to improve the overall accuracy - try multiplying features by overall accuracy for further training instead of feature labels

- ❖ Changing number of decision trees has no effect on Overall Accuracy - Try and find out why and possible solutions

- ❖ Repeat the training using an imbalanced training set and observe results. Further, come up with other measures of accuracy apart from OA since the dataset is imbalance - F-Scores, Kappa coefficients

- ❖ Try out other stacking algorithms and observe changes in the performance

REFERENCES

- Iman Khosravi & Seyed Kazem Alavipanah (2019) A random forest-based framework for crop mapping using temporal, spectral, textural and polarimetric observations, International Journal of Remote Sensing, 40:18, 7221-7251, DOI: 10.1080/01431161.2019.1601285

- https://www.analyticsvidhya.com/blog/2021/04/getting-into-random-forest-algorithms/